Computers and Mathematics with Applications 62 (2011) 3221–3235

Contents lists available at SciVerse ScienceDirect

Computers and Mathematics with Applications

journal homepage: www.elsevier.com/locate/camwa

A class of collocation methods for numerical integration of initial value problems

F. Costabile, A. Napoli *

Department of Mathematics, University of Calabria, 87036 Rende (Cs), Italy

ARTICLE INFO

Article history:
Received 10 June 2010
Received in revised form 10 August 2011
Accepted 10 August 2011

Keywords:
Initial value problem
Collocation methods

ABSTRACT

For the numerical solution of initial value problems a general procedure to determine global integration methods is derived and studied. They are collocation methods which can be easily implemented and provide a high order accuracy. They further provide globally continuous differentiable solutions. Computation of the integrals which appear in the coefficients are generated by a recurrence formula and no integrals are involved in the calculation. Numerical experiments provide favorable comparisons with other existing methods.

1. Introduction

We consider the initial value problem (IVP hereafter) in ordinary differential equations

$$\begin{cases} y'(x) = f(x, y(x)) & x \in [x_0, b] \\ y(x_0) = y_0. \end{cases} \tag{1}$$

We assume that $f(x, y(x))$ satisfies sufficient conditions to guarantee that a unique solution of (1) exists, that is, it is a real function defined and continuous on the strip $S = [a, b] \times \mathbb{R}$ and a constant L exists so that $\forall x \in [a, b]$ and for any two numbers y_1 and y_2

$$|f(x, y_1) - f(x, y_2)| \le L |y_1 - y_2| .$$

Problems of this kind arise in almost all branches of science, engineering and technology, molecular-dynamics calculations for liquid and gases, stellar mechanics and atomic and nuclear scattering problems.

The known methods for the numerical solution of (1) generally fall into two categories: continuous methods which include collocation methods, and discrete methods which include extrapolation, Runge–Kutta and linear multistep methods. Any continuous method can produce approximations at discrete points, but many discrete methods cannot be used to obtain continuous approximations (this is the case of extrapolation and most Runge–Kutta methods). For this reason they are inefficient for problems requiring globally continuous differentiable functions as approximations of $y(x)$.

Here we develop a class of collocation methods, that is methods which produce smooth, global approximations to $y(x)$ in the form of algebraic polynomial functions by requiring that the polynomials satisfy the given differential equation on a suitable finite subset of $[x_0, b]$, and coincide with the exact solution at the initial point y_0. Many authors have studied collocation methods for numerically solving (1) (see, for instance, [1–5], and, more recently [6,7], and references therein). In [2,8] collocation methods for the global approximation of IVPs based on special sets of nodes have been derived and studied. These methods can be considered special cases of the more general methods presented here; this fact motivates the study.

* Corresponding author. Tel.: +39 0984 496488.
E-mail addresses: costabil@unical.it (F. Costabile), a.napoli@unical.it (A. Napoli).

0898-1221/$ – see front matter
doi:10.1016/j.camwa.2011.08.036

We assume that (1) represents a single scalar equation, but nearly all of the numerical and theoretical considerations in this paper carry over systems of first order equations, where (1) could be treated in vector form. Thus, for higher order differential equations we may solve them numerically by first reducing them to systems of first order equations. However, for equations of the form

$$\begin{cases} y^{(k)}(x) = f(x, y(x)) \\ y^{(h)}(x_0) = y_0^h \quad h = 0, \dots, k-1 \end{cases}$$

it is more convenient to attack them directly. In fact, it is well known that several advantages (substantial gain in efficiency, lower storage requirements, etc.) are realized when the equations are treated in their original k-order form [7]. For these reasons, by using the same technique as for first order problems, we also derive collocation methods for solving initial value problems of k-th order, $k > 1$.

The outline of the paper is the following: in Section 2 we present the methods; in Section 3 we study the error and in Section 4 we consider some special cases. Then, in order to improve the performance of the proposed methods, we use piecewise polynomial functions and obtain the corresponding implicit Runge–Kutta methods (Section 5). The use of piecewise polynomials offers significant advantages. For example piecewise polynomial functions are more adaptable to special problems. Moreover it is relatively simple to set up the equations, solve them, vary the order of convergence and adapt the mesh to a particular solution. In Section 6 we consider the case of problems of k order. Then, in Section 7 we present some algorithms for the computation of the polynomials and an algorithm to generate the integration coefficients. Finally, we present the results of some numerical experiments to show that the proposed methods can be competitive with standard methods used in solving stiff and non stiff problems.

2. Collocation methods based on numerical integration

If $y(x)$ is the solution of (1) and $\{x_i\}_{i=1}^n$ are distinct points in $(x_0, b]$, then the following result holds

Proposition 1. *If $y(x) \in C^n[x_0, b]$ and the $(n+1)$-th derivative exists, then*

$$y(x) = y(x_0) + \sum_{i=1}^{n} p_{n,i}(x) f\Big(x_i, y(x_i)\Big) + \frac{1}{n!}\int_{x_0}^{x} \omega_n(t) y^{(n+1)}(\xi_t)\mathrm{d}t \tag{2}$$

where $\xi_t \in (x_0, b)$,

$$p_{ni}(x) = \int_{x_0}^{x} l_i(t)\mathrm{d}t \quad i = 1, \dots, n, \tag{3}$$

$l_i(t),\ i = 1, \dots, n$, are the fundamental Lagrange polynomials on nodes x_i,

$$l_i(t) = \prod_{\substack{k=1 \\ k \neq i}}^{n} \frac{t - x_k}{x_i - x_k}$$

and $\omega_n(t) = (t - x_1) \cdots (t - x_n)$.

Proof. From Lagrange interpolation we have

$$y'(x) = \sum_{i=1}^{n} l_i(x) y'(x_i) + R_n\left(y', x\right) \tag{4}$$

with

$$R_n\left(y', x\right) = \frac{\omega_n(x)}{n!}\left(y'(x)\right)^{(n)}\big|_{x=\xi_x}, \quad \xi_x \in (x_0, b).$$

From (1), (4) and the identity

$$y(x) - y(x_0) = \int_{x_0}^{x} y'(t)\mathrm{d}t$$

we obtain (2). □

We have the following

Theorem 1. *The polynomial of degree n, implicitly defined by*

$$y_n(x) = y_0 + \sum_{i=1}^{n} p_{ni}(x) f\left(x_i, y_n(x_i)\right) \tag{5}$$

satisfies the relations

$$\begin{aligned} & y_n(x_0) = y_0 \\ & y_n'\left(x_j\right) = f\left(x_j, y_n\left(x_j\right)\right), \quad j = 1, \ldots, n \end{aligned} \tag{6}$$

i.e. it is a collocation polynomial for (1) at nodes $x_j,\ j = 1, \ldots, n$.

Proof. For all $i, n \in \mathbb{N}$ polynomials $p_{ni}(x),\ i = 1, \ldots, n$, satisfy

$$p_{ni}(x_0) = 0, \qquad p_{ni}'(x_k) = l_i(x_k) = \delta_{ik}, \quad k = 1, \ldots, n$$

where δ_{ik} is the Kronecker symbol, and this proves (6). □

Remark 1. We explicitly note that the initial value x_0 can also be arbitrarily chosen among the points $\{x_i\}_{i=1}^n$. Furthermore, if $x_i,\ i = 1, \ldots, n$ are equidistant nodes, for an appropriate choice of x_0, x and n, methods (5) coincide with Adams-type methods [1,4].

3. Error estimates

Let

$$T_n(y, x) = y(x) - \left[y_0 + \sum_{i=1}^n p_{ni}(x) y'(x_i)\right] \tag{7}$$

be the truncation error for (5). In the hypothesis of Proposition 1 it holds that

$$T_n(y, x) = \frac{1}{n!} \int_{x_0}^x \omega_n(t) y^{(n+1)}(\xi_t) \mathrm{d}t.$$

Thus, if $M_{n+1} = \max_{x_0 \le x \le b} \left|y^{(n+1)}(x)\right|$ and $N_n = \max_{x_0 \le x \le b} |\omega_n(x)|$,

$$\begin{aligned} |T_n(y, x)| &\le \frac{1}{n!} \int_{x_0}^x |\omega_n(t)| \left|y^{(n+1)}(\xi_t)\right| \mathrm{d}t \\ &\le \frac{M_{n+1} N_n}{n!} (x - x_0). \end{aligned}$$

Now, let's define

$$\Delta_n = \max_{x_0 \le x \le b} \sum_{i=1}^n |p_{ni}(x)| \tag{8}$$

and let $\Lambda_n = \max_{x_0 \le x \le b} \sum_{i=1}^n |l_i(x)|$ be the Lebesgue constant. The following theorem provides an a priori estimate for the truncation error

Theorem 2. *With the notations used above, if L is the Lipschitz constant of f and $L\Delta_n < 1$, then*

$$\|y_n - y\|_\infty \le \frac{M_{n+1} N_n (b - x_0)}{n!(1 - L\Delta_n)} \le \frac{M_{n+1} (b - x_0)^{n+1}}{n!(1 - L\Delta_n)} \tag{9}$$

and

$$\left\|y_n' - y'\right\|_\infty \le \frac{M_{n+1} N_n (b - x_0)}{n!} \left(1 + \frac{L\Lambda_n (b - x_0)}{1 - L\Delta_n}\right). \tag{10}$$

Proof. From (2) and (5) we have

$$y(x) - y_n(x) = \sum_{i=1}^n p_{ni}(x) \Big[f\big(x_i, y(x_i)\big) - f\big(x_i, y_n(x_i)\big)\Big] + \frac{1}{n!} \int_{x_0}^x \omega_n(t) y^{(n+1)}(\xi_t) \,\mathrm{d}t. \tag{11}$$

Considering the absolute value and taking the maximum over all the x this implies (9). By deriving (11) and using (9) we obtain (10). □

Remark 2. It holds that $\Delta_n \le (b - x_0)\, \Lambda_n$.

Remark 3. It is known [9] that if x_i are equidistant nodes in $[-1, 1]$, $N_n = n! \left(\frac{2}{n+1}\right)^n$. In the case of zeros of Chebyshev polynomials of the first kind $N_n = \frac{1}{2^{n-1}}$.

4. Choice of nodes

In the case of particular nodes an explicit expression of polynomial (3) can be obtained.

A. *Zeros of orthogonal polynomials*.
If x_i are the zeros of orthogonal polynomials $\{P_n(x)\}$ in $[-1, 1]$ with respect to a certain weight function $w(x)$, then [10] the fundamental Lagrange polynomials can be written as

$$l_k(x) = \lambda_{nk} \sum_{i=0}^{n-1} P_i(x) P_i(x_k) \tag{12}$$

where λ_{nk} are the coefficients of the Gaussian quadrature formula (Cotes numbers)

$$\int_{-1}^{1} w(x) f(x) \mathrm{d}x = \sum_{k=1}^{n} \lambda_{nk} f(x_k) + R_n(f). \tag{13}$$

Moreover the following theorem holds

Theorem 3. *If x_i are the zeros of orthogonal polynomials $\{P_n(x)\}$ in $[-1, 1]$ with respect to the weight function $w(x)$, then the polynomial (5) has the following form*

$$y_n(x) = y_0 + \sum_{k=1}^{n} \gamma_{nk}(x) f\left(x_k, y_n(x_k)\right)$$

where

$$\gamma_{nk}(x) = \lambda_{nk} \sum_{i=0}^{n-1} P_i(x_k) \int_{-1}^{x} P_i(t) \mathrm{d}t$$

and

$$\lambda_{nk} = \int_{-1}^{1} w(x) \frac{P_n(t)}{(t - x_k) P_n'(x_k)} \mathrm{d}t.$$

Now we consider some particular cases:
Case 1. If x_i, $i = 1, \ldots, n$ are the zeros of normalized Chebyshev polynomials of the first kind

$$P_0(x) = \frac{1}{\sqrt{2\pi}} T_0(x), \qquad P_n(x) = \sqrt{\frac{2}{\pi}} T_n(x) \quad n \geq 1$$

(where $T_n(x) = \cos(n \arccos x)$) that is

$$x_i = \cos \frac{2i - 1}{2n} \pi \quad i = 1, \ldots, n$$

we have that

$$\lambda_{nk} = \frac{\pi}{n} \quad k = 1, \ldots, n.$$

Furthermore, after simple calculation,

$$\int_{-1}^{x} P_0(t) \mathrm{d}t = \frac{x + 1}{\sqrt{2\pi}}$$

$$\int_{-1}^{x} P_1(t) \mathrm{d}t = \sqrt{\frac{2}{\pi}} \left(\frac{x^2}{2} - \frac{1}{2} \right)$$

$$\int_{-1}^{x} P_i(t) \mathrm{d}t = \sqrt{\frac{2}{\pi}} \left\{ \frac{1}{2} \left[\frac{T_{i+1}(x)}{i + 1} - \frac{T_{i-1}(x)}{i - 1} \right] + \frac{(-1)^{i-1}}{i^2 - 1} \right\} \quad i > 1$$

and

$$P_j(x_k) = \sqrt{\frac{2}{\pi}} \cos \left(\frac{2k - 1}{2n} j\pi \right).$$

Thus we have

$$\gamma_{nk}(x) = \frac{1}{n} \sum_{j=2}^{n-1} \left\{ \left[\frac{T_{j+1}(x)}{j + 1} - \frac{T_{j-1}(x)}{j - 1} + 2 \frac{(-1)^{j-1}}{j^2 - 1} \right] \cos \left(\frac{2k - 1}{2n} j\pi \right) \right\} + \frac{1}{n} \left\{ x + 1 + \cos \left(\frac{2k - 1}{2n} \pi \right) \left(x^2 - 1 \right) \right\}.$$

Case 2. If

$$x_i = \cos \frac{\pi i}{n+1} \quad i = 1, \dots, n$$

are the zeros of normalized Chebyshev polynomials of the second kind

$$P_n(x) = \sqrt{\frac{2}{\pi}} U_n(x) \quad n = 0, 1, \dots$$

where $U_n(x) = \frac{\sin((n+1)\arccos x)}{\sqrt{1-x^2}}$, we have that

$$\lambda_{nk} = \frac{\pi}{n+1} \sin^2 \frac{k\pi}{n+1} \quad k = 1, \dots, n.$$

Thus

$$\int_{-1}^{x} P_i(t)\mathrm{d}t = \sqrt{\frac{2}{\pi}} \frac{1}{i+1} \left[T_{i+1}(x) + (-1)^i\right]$$

and

$$P_i(x_k) = \sqrt{\frac{2}{\pi}} \frac{\sin \frac{(i+1)\pi k}{n+1}}{\sin \frac{k\pi}{n+1}} \quad i = 0, 1, \dots.$$

Hence we have

$$\gamma_{nk}(x) = \frac{2}{n+1} \sin \frac{k\pi}{n+1} \sum_{i=1}^{n} \frac{T_i(x) - (-1)^i}{i} \sin \frac{ik\pi}{n+1}.$$

It coincides with the polynomial derived in [2].

B. *Generic nodes*.
If $\{x_i\}_{i=1}^n$ is a set of generic nodes in $[a, b]$, the integral in (3) becomes

$$p_{ni}(x) = \frac{x - x_0}{b - a} \int_a^b l_i\,(x(t))\,\mathrm{d}t \tag{14}$$

where $x(t) = x_0 + \frac{x - x_0}{b-a}(t - a)$. The integral $\int_a^b l_i\,(x(t))\,\mathrm{d}t$ can be computed by a quadrature formula of interpolation type

$$\int_a^b l_i\,(x(t))\,\mathrm{d}t = \sum_{k=1}^{n} A_k l_i(z_k) \tag{15}$$

with $z_k = x_0 + \frac{x - x_0}{b-a}(x_k - a)$.
For example, if $[a, b] = [-1, 1]$ and x_k are the zeros of Legendre polynomials in $[-1, 1]$, we have

$$p_{ni}(x) = \frac{x - x_0}{2} \sum_{k=1}^{n} A_k \prod_{\substack{j=1 \\ j \neq i}}^{n} \frac{\frac{x+x_0}{2} + \frac{x-x_0}{2} x_k - x_j}{x_i - x_j} \quad i = 1, \dots, n$$

where A_k, $k = 1, \dots, n$, are the coefficients of the Gaussian quadrature formula.
If the nodes are equidistant in $[-1, 1]$, $x_i = -1 + \frac{2i}{n}$

$$p_{ni}(x) = \frac{x - x_0}{2} \sum_{k=1}^{n} A_k \prod_{\substack{j=1 \\ j \neq i}}^{n} \frac{kx + (n-k)x_0 + 1 - 2j}{2(i-j)} \quad i = 1, \dots, n$$

where A_k, $k = 1, \dots, n$, are the weights of the Newton–Cotes formula.

5. The corresponding implicit Runge–Kutta methods

It is well known that any one-step collocation method is equivalent to some implicit Runge–Kutta method [1,4]. Therefore we derive the particular Runge–Kutta method to which (5) is equivalent. Of course, collocation methods yield continuous approximation, so "equivalent" here means "matches the same discrete values of the Runge–Kutta approximation".

Let $\chi : t_i = t_0 + ih,\ i = 1, \dots, n$ be a uniform mesh for the sake of simplicity (step size changes are easy, being (5) a one-step method) with $t_0 = x_0$.

On each subinterval $[t_i, t_{i+1}]$ we apply the method (5), so that we have a collocation method

$$y_n(x) = y_i + h \sum_{j=1}^{n} \overline{p}_{nj}(x) f\left(x_j^{(i)}, y_n\left(x_j^{(i)}\right)\right)$$

on the points

$$x_j^{(i)} = t_i + c_j h, \quad \text{with } c_j = \frac{x_j - a}{b - a},\ j = 1, \ldots, n$$

which are the images of the x_j under a linear transform mapping $[x_0, b]$ onto $[t_i, t_{i+1}]$, and

$$\overline{p}_{nj}(x) = \frac{1}{b-a} p_{n,j}\left(\frac{2x - t_{i+1} - t_i}{h}\right).$$

So we have:

$$\begin{cases} y_{i+1} = y_i + h \sum_{j=1}^{n} b_j k_j & i = 0, 1, \ldots \\ k_j = f\left(t_i + c_j h, y_i + h \sum_{m=1}^{n} a_{jm} k_m\right) \end{cases} \tag{16}$$

where

$$b_j = \overline{p}_{nj}\,(t_{i+1}) \qquad a_{jm} = \overline{p}_{nm}\left(x_j^{(i)}\right)$$

$$c_j = \sum_{m=1}^{n} a_{jm}, \quad \text{and} \quad \sum_{j=1}^{n} b_j = 1$$

which is the implicit Runge–Kutta method based on the n-points formula (5).

Being (5) a collocation method on n distinct points, the corresponding implicit Runge–Kutta method (16) has order at least n [4].

In [11] the coefficients have been explicitly calculated for $n = 1, \ldots, 4$ in the case of zeros of Chebyshev polynomials of the second kind.

If $\{x_i\}$ are the zeros of Chebyshev polynomials of the first kind we have the following methods of order, respectively, 2, 2 and 4, which can be expressed in Butcher tableau [1] form (Tables 1–3).

$$\begin{array}{c|c} \frac{1}{2} & \frac{1}{2} \\ \hline & 1 \end{array}$$

Table 1. $n = 1$

$$\begin{array}{c|cc} \frac{2+\sqrt{2}}{4} & \frac{4+\sqrt{2}}{16} & \frac{4+3\sqrt{2}}{16} \\ \frac{2-\sqrt{2}}{4} & \frac{4-3\sqrt{2}}{16} & \frac{4-\sqrt{2}}{16} \\ \hline & \frac{1}{2} & \frac{1}{2} \end{array}$$

Table 2. $n = 2$

$$\begin{array}{c|ccc} \frac{2+\sqrt{3}}{4} & \frac{1}{9} + \frac{1}{16\sqrt{3}} & \frac{5+3\sqrt{3}}{18} & \frac{1}{9} + \frac{\sqrt{3}}{16} \\ \frac{1}{2} & \frac{4-3\sqrt{3}}{36} & \frac{5}{18} & \frac{4+3\sqrt{3}}{36} \\ \frac{2-\sqrt{3}}{4} & \frac{1}{9} - \frac{\sqrt{3}}{16} & \frac{5-3\sqrt{3}}{18} & \frac{1}{9} - \frac{1}{16\sqrt{3}} \\ \hline & \frac{2}{9} & \frac{5}{9} & \frac{2}{9} \end{array}$$

Table 3. $n = 3$

If $\{x_i\}$ are equidistant nodes in $[-1, 1]$ we have the following methods (Tables 4–6) of order, respectively, 1, 2 and 3.

$$\begin{array}{c|c} 1 & 1 \\ \hline & 1 \end{array}$$

Table 4. $n = 1$

$$\begin{array}{c|cc} \frac{1}{2} & \frac{3}{4} & -\frac{1}{4} \\ 1 & 1 & 0 \\ \hline & 1 & 0 \end{array}$$

Table 5. $n = 2$

$$\begin{array}{c|ccc} \frac{1}{3} & \frac{23}{36} & -\frac{4}{9} & \frac{5}{36} \\ \frac{2}{3} & \frac{7}{9} & -\frac{2}{9} & \frac{1}{9} \\ 1 & \frac{3}{4} & 0 & \frac{1}{4} \\ \hline & \frac{3}{4} & 0 & \frac{1}{4} \end{array}$$

Table 6. $n = 3$

Note that method in Table 1 is the implicit midpoint rule and the method in Table 4 is the implicit Euler method.

Stability regions of these methods can be calculated using classical techniques and are drawn in Figs. 1 and 2. We can observe that these methods are A-stable or have a large stability region.

5.1. Relationship with Galerkin methods

Using (12), the following theorem may be proved with standard techniques

Theorem 4. *Let $\{x_i\}_{i=1}^{n}$ be the zeros of orthogonal polynomials $P_n(x)$ in $[-1, 1]$ with respect to a certain weight function $w(x)$. If we approximate the inner product*

$$(u, v) = \int_{-1}^{1} w(x)u(x)v(x)\mathrm{d}x$$

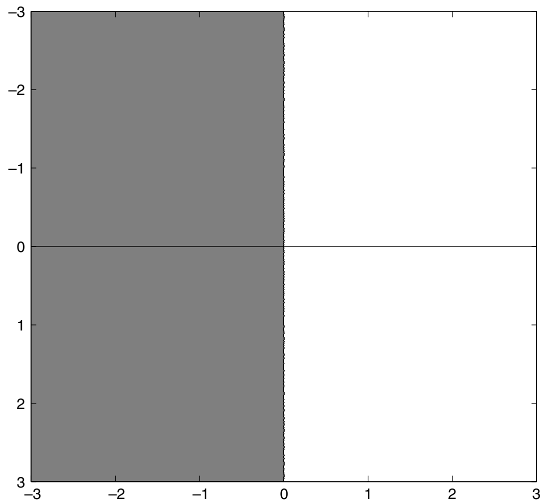

Fig. 1. Stability regions—Chebyshev nodes of the first kind $n = 1, 2, 3$.

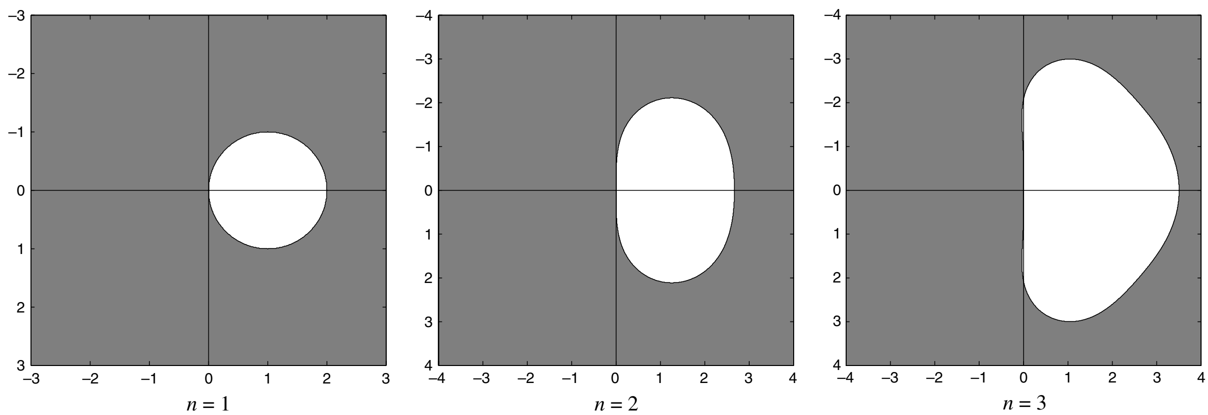

Fig. 2. Stability regions—equidistant nodes.

by the discrete inner product given by Gaussian quadrature formula (13),

$$(u, v) = \int_{-1}^{1} w(x)u(x)v(x)\mathrm{d}x \doteq \sum_{k=1}^{n} \lambda_{nk} u(x_k)v(x_k), \tag{17}$$

then the collocation method (5) on $\{x_i\}$ is a Galerkin-type method and vice versa.

6. Initial value problems of order $k > 1$

With the same technique used above we can derive collocation methods for the solution of initial value problems of order $k > 1$

$$\begin{cases} y^{(k)}(x) = f(x, y(x)) \\ y^{(h)}(x_0) = y_0^h \quad h = 0, \ldots, k-1. \end{cases} \tag{18}$$

Let us suppose that $y(x)$ is the solution of (18) and $\{x_i\}_{i=1}^{n}$ are distinct points in (x_0, b). Then the following result holds

Proposition 2. *If $y(x) \in C^{n+k-1}[x_0, b]$ and the $(n+k)$-th derivative exists, then*

$$y(x) = \sum_{i=0}^{k-1} \frac{(x-x_0)^i}{i!} y_0^i + \sum_{i=1}^{n} q_{ni}(x) f(x_i, y(x_i)) + R_n(f, x) \tag{19}$$

where

$$q_{ni}(x) = \underbrace{\int_{x_0}^{x} \cdots \int_{x_0}^{x}}_{k} l_i(t)\mathrm{d}t \cdots \mathrm{d}t, \tag{20}$$

$$R_n(f, x) = \frac{1}{n!} \underbrace{\int_{x_0}^{x} \cdots \int_{x_0}^{x}}_{k} \omega_n(t) y^{(n+k)}(\xi_t)\mathrm{d}t \cdots \mathrm{d}t.$$

Proof. From (18) we have

$$y(x) = \sum_{i=0}^{k-1} \frac{(x-x_0)^i}{i!} y_0^i + \underbrace{\int_{x_0}^{x} \cdots \int_{x_0}^{x}}_{k} f(t, y(t))\mathrm{d}t \cdots \mathrm{d}t. \tag{21}$$

From Lagrange interpolation on the nodes $\{x_i\}_{i=1}^n$

$$f(x, y(x)) = \sum_{i=1}^{n} l_i(x) f(x_i, y(x_i)) + \frac{\omega_n(x)}{n!} y^{(n+k)}(\xi_x).$$

By substituting in (21) the result follows.

It's easy to prove the following results. □

Theorem 5. *The implicitly defined polynomial of degree n*

$$y_n(x) = \sum_{i=0}^{k-1} \frac{(x-x_0)^i}{i!} y_0^i + \sum_{i=1}^{n} q_{ni}(x) f(x_i, y_n(x_i)) \tag{22}$$

satisfies the relations

$$\begin{aligned} &y_n^{(h)}(x_0) = y_0^h, \quad h = 0, \dots, k-1 \\ &y_n^{(k)}\left(x_j\right) = f\left(x_j, y_n\left(x_j\right)\right), \quad j = 1, \dots, n \end{aligned} \tag{23}$$

i.e. it is a collocation polynomial for (18) *at nodes* x_j, $j = 1, \dots, n$.

For example, for $k = 2$, if $\{x_i\}_{i=1}^n$ are the zeros of Chebyshev polynomials of the first and second kind, we have, respectively, the following explicit expressions for polynomials $q_{ni}(x)$

$$\begin{aligned} q_{ni}(x) = \frac{1}{n} \Bigg\{ & \frac{(x+1)^2}{2} + \frac{x^3 - 3x - 2}{3} \left(\cos\frac{\pi(2i-1)}{2n} + x\cos\frac{\pi(2i-1)}{n} \right) + \frac{1}{2} \sum_{k=3}^{n-1} \cos\frac{k\pi(2i-1)}{2n} \\ & \times \left[\frac{T_{k+2}(x)}{(k+1)(k+2)} - 2\frac{T_k(x)}{k^2-1} + \frac{T_{k-2}(x)}{(k-1)(k-2)} - \frac{12k(-1)^k}{k(k^2-1)(k^2-4)} - \frac{4(-1)^k}{k^2-1}(x+1) \right] \Bigg\} \end{aligned} \tag{24}$$

and

$$q_{ni}(x) = \frac{1}{n+1} \sin\frac{\pi i}{n+1} \left\{ \sin\frac{\pi i}{n+1} (x+1)^2 + \sum_{k=2}^{n} \frac{1}{k} \sin\frac{k\pi i}{n+1} \left[\frac{T_{k+1}(x)}{k+1} - \frac{T_{k-1}(x)}{k-1} - 2\left(x + \frac{k^2}{k^2-1}\right)(-1)^k \right] \right\}. \tag{25}$$

In [12] Coleman and Booth, starting from the Panovsky–Richardson method [13], and by using a different identity from (21), derived a collocation method based on the polynomial interpolant of degree n for y'', for which the nodes $\{x_i\}_{i=1}^n$ are the zeros of Chebyshev polynomials of the second kind.

In the case of $x_i = -\cos\frac{\pi i}{n+1}$, polynomial (22) corresponds to the collocation method introduced in [8].

For the truncation error

$$T_n(y, x) = y(x) - \left[\sum_{i=0}^{k-1} \frac{(x-x_0)^i}{i!} y_0^i + \sum_{i=1}^{n} q_{ni}(x) y''(x_i) \right].$$

From Proposition 2, if $M_{n+k} = \max_{x_0 \le x \le b} \left|y^{(n+k)}(x)\right|$ and $N_n = \max_{x_0 \le x \le b} |\omega_n(x)|$,

$$|T_n(y,x)| \le \frac{1}{n!} \underbrace{\int_{x_0}^{x} \cdots \int_{x_0}^{x}}_{k} |\omega_n(t)| \left|y^{(n+k)}(\xi_t)\right| \mathrm{d}t \cdots \mathrm{d}t$$

$$\le M_{n+k} N_n \frac{(x-x_0)^k}{n!k!}.$$

Theorem 6. *If* $\overline{\Delta}_{nj} = \max_{x_0 \le x \le b} \sum_{i=1}^{n} \left|q_{ni}^{(j)}(x)\right|$ $j = 1, \dots, k$ *and* $L\overline{\Delta}_{n0} < 1$, *with the notations used above, we have*

$$\left\|y_n^{(s)} - y^{(s)}\right\|_\infty \le M_{n+k} N_n \frac{(b-x_0)^{k-s}}{n!} G_{ks} \quad s = 0, 1, \dots, k$$

where

$$G_{ks} = \begin{cases} \dfrac{1}{k!(1-L\overline{\Delta}_{n0})} & s = 0 \\ \dfrac{1}{(k-s)!} + \dfrac{L\overline{\Delta}_{ns}\,(b-x_0)^s}{k!(1-L\overline{\Delta}_{n0})} & s = 1, \dots, k. \end{cases}$$

6.1. Continuous Runge–Kutta methods

Like in the case of first order equations, for each method (22) we can derive the corresponding implicit Runge–Kutta method. For example, for $k = 2$, let $b = x_0 + h$ and $x_i = x_0 + c_i h$ with $c_i \in [0, 1]$. With the change of coordinates $x = x_0 + th$, $t \in [0, 1]$, we can write

$$q_{ni}(x) = q_{ni}(x_0 + th) = h^2 \int_0^t \int_0^r l_i(s)\mathrm{d}s\mathrm{d}r, \quad l_i(s) = \prod_{\substack{k=1 \\ k \ne i}}^{n} \frac{s - c_k}{c_i - c_k}.$$

Putting

$$f\left(x_i, y_n(x_i)\right) = y_n''(x_i) \equiv K_i, \qquad a_{i,j} = q_{nj}(x_i) = h^2 \int_0^{c_i} (c_i - s) l_j(s)\mathrm{d}s,$$

we have

$$K_i = f\left(x_0 + c_i h, y_0 + y_0' th + \sum_{j=1}^{n} a_{i,j} K_j\right) \tag{26}$$

and

$$\begin{cases} y_1 \equiv y_n(x_0 + th) = y_0 + y_0' th + h^2 \displaystyle\sum_{i=1}^{n} q_{ni}(x_0 + th) K_i \\ y_1' \equiv y_n'(x_0 + th) = y_0' h + h^2 \displaystyle\sum_{i=1}^{n} q_{ni}'(x_0 + th) K_i. \end{cases} \tag{27}$$

(26)–(27) is the well known *continuous Runge–Kutta* method [4] for second order differential equations. Particularly, for $t = 1$ we have the implicit Runge–Kutta–Nystrom method [1,12,4].

7. Algorithms and implementation

In order to calculate an approximate solution of (1) we need the values $y_n(x_r)$, $r = 1, \dots, n$. For this aim we can solve the system

$$y_n(x_r) = y_0 + \sum_{i=1}^{n} p_{ni}(x_r) f\left(x_i, y_n(x_i)\right) \quad r = 1, \dots, n. \tag{28}$$

Putting

$$Y_n = [y_n(x_1), \dots, y_n(x_n)]^T, \qquad F(Y_n) = \left[f\left(x_1, y_n(x_1)\right), \dots, f\left(x_n, y_n(x_n)\right)\right]^T$$

$$A \equiv \left(a_{ij}\right) = \left(p_{nj}(x_i)\right), \qquad Y_0 = [y_0, \dots, y_0]^T$$

the system can be written in the following form

$$Y_n - AF(Y_n) = Y_0. \tag{29}$$

In general (28) can be solved by an iterative method

$$Y_n^{(\nu+1)} = AF\left(Y_n^{(\nu)}\right) + Y_0, \quad \nu = 1, \ldots, N \tag{30}$$

with $Y_n^{(0)}$ being a starting value. If

$$G(Y) = AF(Y) + Y_0,$$

(29) and (30) become respectively

$$Y_n = G(Y_n) \tag{31}$$

and

$$Y_n^{(\nu+1)} = G\left(Y_n^{(\nu)}\right), \quad \nu = 1, \ldots, N. \tag{32}$$

Then, for each Y_n^1 and Y_n^2 we have

$$G\left(Y_n^1\right) - G\left(Y_n^2\right) = A\left[F\left(Y_n^1\right) - F\left(Y_n^2\right)\right]$$

and

$$\left\|G\left(Y_n^1\right) - G\left(Y_n^2\right)\right\| \le \|A\| L \left\|Y_n^1 - Y_n^2\right\|.$$

Therefore, if $\|A\|\, L < 1$, with L the Lipschitz constant of f, G is contractive and (31) has a unique solution to which iterates (32) converge.

Iterations (32) correspond to the approximate computations of Picard iterations for (1). In fact, from

$$y^{(\nu+1)}(x) = y_0 + \int_{x_0}^{x} f\left(t, y^{(\nu)}(t)\right) \mathrm{d}t \tag{33}$$

if we approximate f by Lagrange polynomial on the nodes $\{x_i\}$, that is

$$f\left(t, y^{(\nu)}(t)\right) = \sum_{i=1}^{n} l_i(t) f\left(x_i, y^{(\nu)}(x_i)\right),$$

we have

$$\begin{aligned} y^{(\nu+1)}(x) &= y_0 + \sum_{i=1}^{n} f\left(x_i, y^{(\nu)}(x_i)\right) \int_{x_0}^{x} l_i(t) \mathrm{d}t \\ &= y_0 + \sum_{i=1}^{n} p_{ni}(x) f\left(x_i, y^{(\nu)}(x_i)\right) \end{aligned}$$

which, for $x = x_i,\ i = 1, \ldots, n$, coincides with (32).

For the computation of (22) we can proceed as in the case of first order problems.

In fact if

$$\overline{A} \equiv \left(\overline{a}_{ij}\right) = \left(q_{nj}\left(x_i\right)\right), \qquad \overline{Y}_0 = \left[\sum_{i=0}^{k-1} \frac{(x_1 - x_0)^i}{i!} y_0^i, \ldots, \sum_{i=0}^{k-1} \frac{(x_n - x_0)^i}{i!} y_0^i\right]^T$$

we can solve the system

$$Y_n - \overline{A} F\left(Y_n\right) = \overline{Y}_0 \tag{34}$$

by an iterative method

$$Y_n^{(\nu+1)} = \overline{A} F\left(Y_n^{(\nu)}\right) + \overline{Y}_0$$

with $\overline{Y}_n^{(0)}$ an initial value.

7.0.1. Calculation of coefficients $p_{ni}(x_k)$ and $q_{ni}(x_k)$

We have seen that for some special set of nodes we have an explicit expression of polynomials $p_{ni}(x)$ and $q_{ni}(x)$, and this allows the computation respectively of the elements of A and of $\overline{A}$. When collocation nodes are arbitrarily taken, the question arises as to how to calculate $p_{ni}(x)$ and $q_{ni}(x)$, that is, how to calculate

$$\int_a^x r_{n,i}(t) \mathrm{d}t \quad \text{and} \quad \underbrace{\int_a^x \cdots \int_a^x}_{k} r_{n,i}(t) \mathrm{d}t \cdots \mathrm{d}t \tag{35}$$

where

$$r_{n,i}(t) = (t - x_1)\cdots(t - x_{i-1})(t - x_{i+1})\cdots(t - x_n) \quad i = 1, 2, \ldots, n$$
$$r_{0,0}(t) = 1 \tag{36}$$

without computing integrals or using quadrature formulas as in [9].

Following the idea in [14], we propose an algorithm to compute (35).

For each i and $m = 1, \ldots, n-1$, let us define

$$g_{m,1}^{(i)}(x) = \int_a^x \left(t - z_1^{(i)}\right)\left(t - z_2^{(i)}\right)\cdots\left(t - z_m^{(i)}\right) \mathrm{d}t, \tag{37}$$

$$g_{0,1}^{(i)}(x) = x - a \tag{38}$$

and the new points $z_j^{(i)}$ such that

$$z_j^{(i)} = \begin{cases} x_j & \text{if } j < i \\ x_{j+1} & \text{if } j \geq i \end{cases} \quad j = 1, \ldots, n-1.$$

Moreover let us define

$$g_{m,j}^{(i)}(x) = \int_a^x \underbrace{\int_a^x \cdots \int_a^x}_{j-1} \left(t - z_1^{(i)}\right)\left(t - z_2^{(i)}\right)\cdots\left(t - z_m^{(i)}\right) \mathrm{d}t \cdots \mathrm{d}t. \tag{39}$$

We can easily compute $g_{1,j}^{(i)}(x)$. In fact

$$g_{1,j}^{(i)}(x) = \frac{\left(x - z_1^{(i)}\right)^{j+1}}{(j+1)!} - \sum_{i=0}^{j-1} \frac{\left(a - z_1^{(i)}\right)^{j+1-i}}{(j+1-i)!}\frac{(x-a)^i}{i!}. \tag{40}$$

Thus, for the computation of (39) the following recurrence formula holds

$$g_{m,j}^{(i)}(x) = \left(x - z_m^{(i)}\right) g_{m-1,j}^{(i)}(x) - j g_{m-1,j+1}^{(i)}(x). \tag{41}$$

The relation (41) can be proved by induction on j. In fact, if $j = 1$, let $u = t - z_m^{(i)}$ and $\mathrm{d}v = \left(t - z_1^{(i)}\right)\left(t - z_2^{(i)}\right)\cdots\left(t - z_{m-1}^{(i)}\right)$. Solving (37) using integration by parts yields

$$\begin{aligned} g_{m,1}^{(i)}(x) &= \left(x - z_m^{(i)}\right)\int_a^x \left(t - z_1^{(i)}\right)\left(t - z_2^{(i)}\right)\cdots\left(t - z_{m-1}^{(i)}\right)\mathrm{d}t - \int_a^x\int_a^s \left(t - z_1^{(i)}\right)\left(t - z_2^{(i)}\right)\cdots\left(t - z_{m-1}^{(i)}\right)\mathrm{d}t\mathrm{d}s \\ &= \left(x - z_m^{(i)}\right) g_{m-1,1}^{(i)}(x) - g_{m-1,2}^{(i)}(x). \end{aligned}$$

If we suppose that the formula is true up to $j-1$, then

$$\begin{aligned} g_{m,j}^{(i)}(x) &= \int_a^x g_{m,j-1}^{(i)}(t)\mathrm{d}t \\ &= \int_a^x \left(t - z_m^{(i)}\right) g_{m-1,j-1}^{(i)}(t)\mathrm{d}t - (j-1)\int_a^x g_{m-1,j}^{(i)}(t)\mathrm{d}t \\ &= \left(x - z_m^{(i)}\right)\int_a^x g_{m-1,j-1}^{(i)}(t)\mathrm{d}t - \int_a^x g_{m-1,j}^{(i)}(t)\mathrm{d}t - (j-1)\int_a^x g_{m-1,j}^{(i)}(t)\mathrm{d}t \\ &= \left(x - z_m^{(i)}\right) g_{m-1,j}^{(i)}(x) - j g_{m-1,j+1}^{(i)}(x). \end{aligned}$$

Thus, if $W_i = \prod_{\substack{j=1 \\ j\neq i}}^n \left(x_i - x_j\right)$,

$$a_{ij} = p_{ni}(x_j) = \frac{g_{n-1,1}^{(i)}\left(x_j\right)}{W_i} \tag{42}$$

and

$$\bar{a}_{ij} = q_{ni}(x_j) = \frac{g_{n-1,2}^{(i)}\left(x_j\right)}{W_i}. \tag{43}$$

8. Numerical examples

Now we present some numerical results obtained by applying methods (5) and (22) to find numerical approximations of the solutions of some test problems. The nodes x_i are

Table 1
Problem (44)— Example 1.

Method	Error	Time (s)
Cheb I	$1.71 \cdot 10^{-9}$	0.039
Cheb II	$1.17 \cdot 10^{-9}$	0.040
EqPts	$1.49 \cdot 10^{-7}$	0.080
Lobatto IIIA	$4.63 \cdot 10^{-9}$	0.025
ode45	$1.14 \cdot 10^{-6}$	0.096

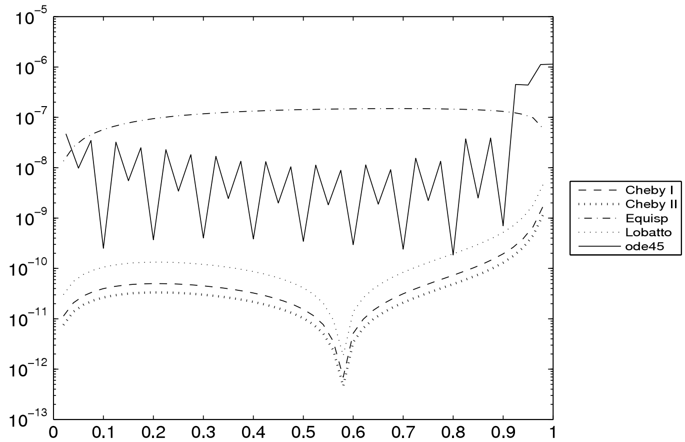

Fig. 3. Example 1. Error functions of problem (44).

- the zeros of Chebyshev polynomials of the first kind (Cheb I)
- the zeros of Chebyshev polynomials of the second kind (Cheb II)
- equidistant points in $[x_0, b]$ (EqPts).

To solve the nonlinear systems (29) and (34) we used the so-called modified Newton method [15] and calculated coefficients by (42) and (43). All problems are classical and discussed in literature (see, e.g., [12,11] and references therein). As the true solutions are known, we considered the error functions $e(x) = |y(x) - y_n(x)|$ and we computed the maximum error in $[x_0, b]$. It will be shown that the proposed methods compare favorably with other existing methods.

Particularly, in Examples 1–3, results are compared with the ones obtained by applying the 3-stage 4-order Lobatto IIIA method [4] and the MatLab solver `ode45` (or `ode15s` in the case of stiff problems). We used piecewise approximation for both methods (5) and Lobatto IIIA, with the same step size $h = 0.02$ and $n = 3$ in (5) in order to have the same number of function evaluations. For each method we give an estimate of the cost in terms of execution time.

For second order problems (Example 4) we compare methods (22) (with $k = 2$) with the Coleman and Booth method [12], which we indicate by the *CB* method. Methods (22) and *CB* have the same cost.

In Example 5 we consider a fifth order initial value problem. In this case method (22) for $k = 5$ is compared with the MatLab solver `ode45`.

Example 1.

$$\begin{cases} y' = -(1-x)^{\frac{3}{2}}\, y & x \in [0, 1] \\ y(0) = 1 \end{cases} \tag{44}$$

with solution $y(x) = e^{\frac{2}{5}\left[(1-x)^{\frac{5}{2}} - 1\right]}$.

The maximum absolute errors on the interval [0, 1] and the execution time for each method are displayed in Table 1. Fig. 3 illustrates the error functions.

Example 2.

$$\begin{cases} y' = 100\,(\sin x - y) & x \in [0, 1] \\ y(0) = 0 \end{cases} \tag{45}$$

with solution $y(x) = \frac{\sin x - 0.01 \cos x + 0.01 e^{-100x}}{1.0001}$.

In Table 2 we compare the maximum absolute errors and in Fig. 4 the graphs of the error functions in [0, 1].

Table 2
Problem (45)— Example 2.

Method	Error	Time (s)
Cheb I	$2.59 \cdot 10^{-5}$	0.010
Cheb II	$2.00 \cdot 10^{-5}$	0.008
EqPts	$4.45 \cdot 10^{-5}$	0.008
Lobatto IIIA	$7.52 \cdot 10^{-5}$	0.007
ode15s	$8.82 \cdot 10^{-6}$	0.242

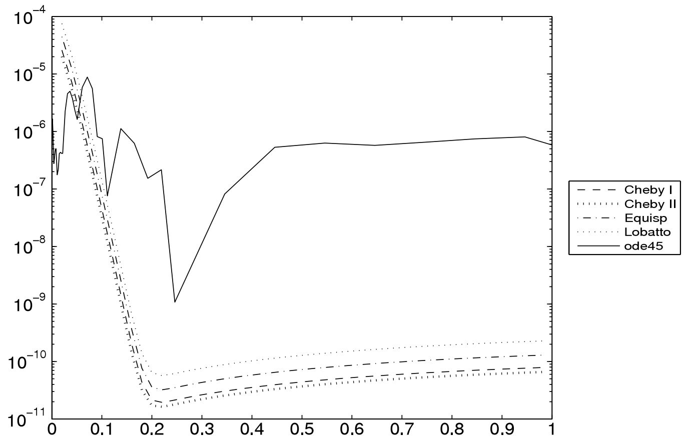

Fig. 4. Example 2. Error functions of problem (45).

Table 3
Problem (46)— Example 3.

Method	Error 1	Error 2	Time (s)
Cheb I	$2.44 \cdot 10^{-9}$	$7.49 \cdot 10^{-9}$	0.053
Cheb II	$1.63 \cdot 10^{-9}$	$4.99 \cdot 10^{-9}$	0.058
EqPts	$3.74 \cdot 10^{-7}$	$1.05 \cdot 10^{-6}$	0.034
Lobatto IIIA	$6.52 \cdot 10^{-9}$	$1.99 \cdot 10^{-8}$	0.073
ode45	$1.24 \cdot 10^{-7}$	$3.74 \cdot 10^{-7}$	0.424

Example 3.

$$\begin{cases} y' = -z \\ z' = -3y - 2z \\ y(0) = 2 \qquad z(0) = 2 \end{cases} \tag{46}$$

with solution $y(x) = e^x + e^{-3x}$, $z(x) = 3e^{-3x} - e^x$.

In Table 3 we compare the maximum absolute errors in $[0, 1]$. Figs. 5 and 6 illustrate the graphs of the error functions in $[0, 1]$ and Fig. 7 illustrate the graphs of the error functions in $[0, 10]$.

Example 4. A non-linear example frequently used in testing numerical methods is provided by the two-body problem:

$$\begin{cases} y'' + \dfrac{y}{r^3} = 0, \quad y(0) = 1 - e, \quad y'(0) = 0 \\ z'' + \dfrac{z}{r^3} = 0, \quad z(0) = 0, \qquad z'(0) = \sqrt{\dfrac{1+e}{1-e}} \end{cases} \tag{47}$$

with $r^2 = y^2 + z^2$. The exact solution is

$$y = \cos E - e, \qquad z = \sqrt{1 - e^2} \sin E,$$

where e is the eccentricity of the orbit and E, the eccentric anomaly, is implicitly defined by Kepler'e equation $x = E - e \sin E$.

Let's call *CBPR* the implicit Panovsky–Richardson method modified by Coleman and Booth [12], which has order $n + 1$ for odd n, and $n + 2$ for even n.

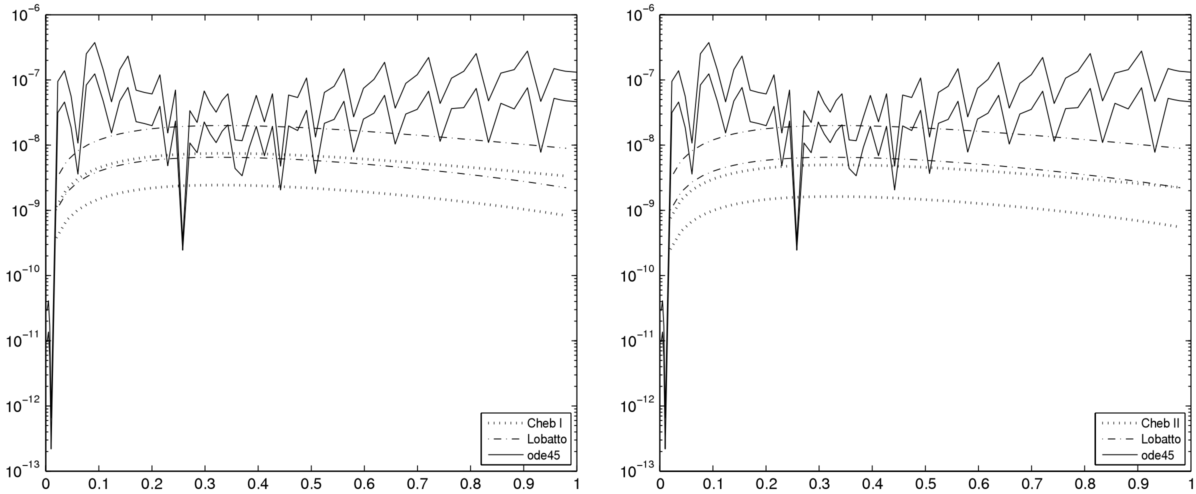

Fig. 5. Example 3. Error functions of problem (46)—Cheb I and Cheb II.

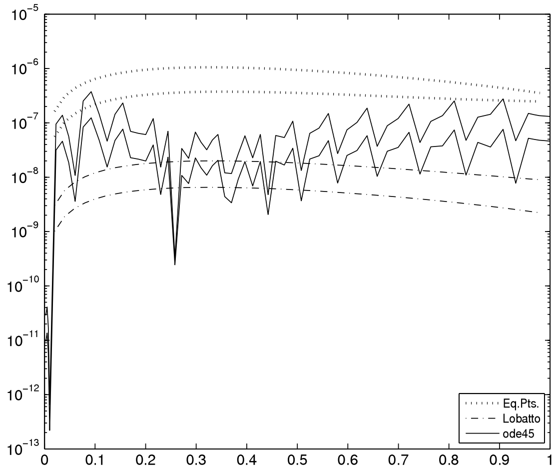

Fig. 6. Example 3. Error functions of problem (46)—Eq.Pts.

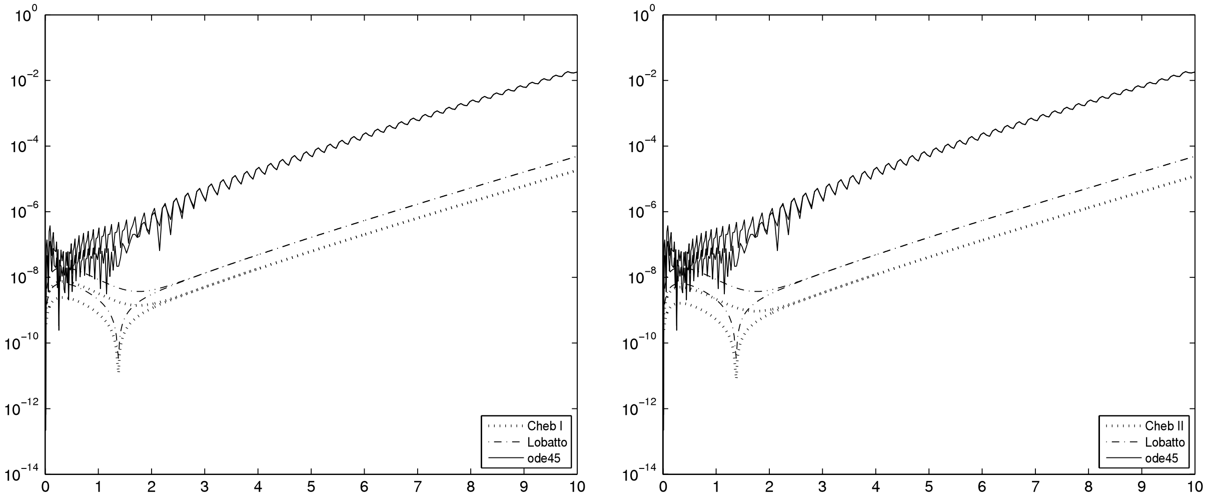

Fig. 7. Example 3. Error functions of problem (46)—Cheb I and Cheb II.

Table 4
Problem (47)— Example 4.

x	Cheb I	Cheb II	EqPts	*CBPR*
1	$6.42 \cdot 10^{-11}$	$4.28 \cdot 10^{-11}$	$9.02 \cdot 10^{-5}$	$5.92 \cdot 10^{-11}$
5	$7.31 \cdot 10^{-10}$	$4.69 \cdot 10^{-10}$	$1.39 \cdot 10^{-3}$	$1.66 \cdot 10^{-9}$
10	$1.38 \cdot 10^{-9}$	$9.20 \cdot 10^{-10}$	$5.12 \cdot 10^{-2}$	$2.15 \cdot 10^{-9}$

Table 5
Problem (48)— Example 5.

Method	Error	Time (s)
Cheb I	$2.89 \cdot 10^{-10}$	0.635
Cheb II	$5.21 \cdot 10^{-10}$	0.883
EqPts	$2.82 \cdot 10^{-10}$	1.778
ode45	$3.01 \cdot 10^{-10}$	0.919

Table 4 compares the maximum absolute errors on $[0, x]$ for the proposed methods in the case respectively of Chebyshev nodes of the first kind, of the second kind and in the case of equidistant nodes, and the *CBPR* method, applied to problem (47) when $e = 0.1$ and steplength $h = 0.02$.

Example 5.

$$\begin{cases} y^{(5)} + \left(32x^5 + 120x\right) y = 160x^3 e^{-x^2} \quad 0 < x < 1 \\ y(0) = 1, \qquad y'(0) = 0, \qquad y''(0) = -2 \\ y'''(0) = 0, \qquad y^{\mathrm{IV}}(0) = 12 \end{cases} \tag{48}$$

with solution $y(x) = e^{-x^2}$.

Table 5 compares the maximum absolute errors on $[0, 1]$ and the execution time for each method.

9. Conclusions

This paper presents a general procedure to determine collocation methods for initial value problems of k-th ($k > 0$) order. For each positive integer n a polynomial approximating the solution is given explicitly. Numerical experiments support theoretical results. Particularly, the zeros of Chebyshev polynomials of the first and second kind provide performances which are highly competitive with other existing methods. Further developments can be done, concerning particularly numerical estimates of the error and the construction of automatic codes.

References

[1] J.C. Butcher, The Numerical Analysis of Ordinary Differential Equations, Runge–Kutta and General Linear Methods, Whiley, Chichester, 1987.
[2] F. Costabile, A. Napoli, A method for global approximation of the initial value problem, Numer. Algorithms 27 (2001) 119–130.
[3] C. de Boor, B. Swartz, Collocation at Gaussian points, SIAM J. Numer. Anal. 10 (4) (1973) 582–606.
[4] E. Hairer, S.P. Norsett, G. Wanner, Solving Ordinary Differential Equations I, Springer-Verlag, Berlin, 1992.
[5] K. Wright, Chebyshev collocation methods for ordinary differential equations, Comput. J. 6 (1963/64) 358–363.
[6] B.S. Ogundare, On the pseudo-spectral methods of solving linear ordinary differential equations, J. Math. Stat. 5 (2) (2009) 136–140.
[7] Y.A. Yahala, A.M. Badmus, A class of collocation methods for general second order ordinary differential equations, Afr. J. Math. Comput. Sci. Res. 2 (4) (2009) 69–72.
[8] F. Costabile, A. Napoli, A method for global approximation of second order IVPs, Rend. Mat. Ser. II 24 (2004) 239–260.
[9] E. Rofman, Osservazioni sulla convergenza di un metodo d'integrazione per punti di equazioni differenziali, Accad. Naz. Lincei Rend. Cl. Sci. Fis. Mat. Natur. Ser. VIII XXXVII (3–4) (1964).
[10] G. Mastroianni, G.V. Milovanovic, Interpolation Processes, Springer, 2008.
[11] F. Costabile, A. Napoli, Stability of Chebyshev collocation methods, Comput. Math. Appl. 47 (2004) 659–666.
[12] J.P. Coleman, A.S. Booth, The Chebyshev methods of Panovsky and Richardson as Runge–Kutta–Nyström methods, J. Comput. Appl. Math. 61 (1995) 245–261.
[13] J. Panovsky, D.L. Richardson, A family of implicit Chebyshev methods for the numerical integration of second-order differential equations, J. Comput. Appl. Math. 23 (1) (1988) 35–51.
[14] Z.B. Omar, M. Suleiman, Solving first order system of ordinary differential equations using parallel r-point block method of variable step size and order, Chiang Mai J. Sci. 36 (1) (2009) 9–23.
[15] A. Quarteroni, R. Sacco, F. Saleri, Numerical Mathematics, Springer-Verlag, New York, 2000.